สุทธิศักดิ์ พงศ์ธนาพาณิช

Programmers's Notes for VB/Win ตอนที่ 4

เทคนิคพิเศษสำหรับผู้อ่านที่สนใจการเขียนโปรแกรมด้วย VB/Win

ในฉบับนี้ยังคงเป็นการรวบรวมเทคนิคพิเศษและคำถามจากผู้อ่านที่จดหมายมายังตู้ ป.ณ. ของผู้เขียน สำหรับผู้อ่านที่สนใจในการเขียนโปรแกรมด้วย VB/Win สามารถนำไปประยุกต์ใช้ได้อย่างรวดเร็ว

การเล่นไฟล์ .AVI ด้วยฟังก์ชัน API

นอกจากผู้อ่านจะใช้คอนโทรล MCI ในการเล่นไฟล์ .avi ได้แล้ว ผู้อ่านยังสามารถใช้ฟังก์ชันวินโดว์ API ที่ชื่อ mciSendString เล่นไฟล์มัลติมีเดียได้อีกด้วย ซึ่งข้อดีของการใช้ฟังก์ชันดังกล่าวก็คือ ผู้อ่านสามารถเขียนโค้ดเพียงบรรทัดเดียวเท่านั้น ซึ่งสามารถประหยัดพื้นที่บนดิสก์สำหรับไฟล์คอนโทรล MCI.VBX ได้อีกด้วย ในตัวอย่างต่อไปนี้ผู้อ่านสามารถกำหนดชื่อไฟล์ .avi ที่ต้องการให้กับรูทีน FullScreenPlayAVI เพื่อเล่นไฟล์มัลติมีเดียได้ทันที

```
Declare Function mciSer
  "mmsystem" (ByVal lps
  lpstrReturnStr As Any
  wReturnLen%, ByVal hC

Sub FullScreenPlayAVI
  Dim cmdStr$, RetVal&

  cmdStr$ =  "play " & 
             " fullscre
  RetVal& = mciSendStri
End Sub
```

หมายเหตุ วินโดว์จะสร้างฟอร์
เพื่อใช้แสดงเล่นไฟล์ .avi และท
ทันทีที่สิ้นสุดการเล่นไฟล์ .avi
ไฟล์ .avi ผู้อ่านสามารถเปลี่ย
ใช้เม้าส์

การเล่นไฟล์ .W
ด้วยฟังก์ชัน AP

ผู้อ่านสามารถเล่นกลับ
วินโดว์ API ที่ชื่อ sndPlaySou
จะทำหน้าที่เล่นกลับไฟล์ .wav
WAVfilename$

```
Declare Function sndP
  "MMSYSTEM.DLL" (ByV
  ByVal wFlags%) As In

Sub PlayWAV (WAVfiler
  Const SND_ASYNC = &
  Const SND_NODEFAULT = &H2

  Dim Dummy%

  Dummy% = sndPlaySound(WAVfilename$,
           SND_ASYNC Or SND_NODEFAULT)
End Sub
```

หมายเหตุ ถ้าหากไฟล์ .wav มีขนาดใหญ่มาก ๆ ฟังก์ชัน sndPlaySound จะไม่สามารถเล่นไฟล์ดังกล่าวได้ แต่ผู้อ่านสามารถอ่านเทคนิคที่สูงกว่านี้ได้จากบทความ "เปิดโลกมัลติมีเดีย" ในนิตยสารไมโครคอมพิวเตอร์

การเล่นไฟล์ .MID ด้วยฟังก์ชัน API

การเล่นไฟล์ .mid นั้นมีเทคนิคที่แตกต่างจากการเล่นไฟล์ .wav ตรงที่คำสั่งที่สนับสนุนและวิธีการอ้างอิงถึงดีไวซ์สำหรับการเล่นไฟล์ .mid แต่ละไฟล์ ซึ่งผู้อ่านสามารถเล่นกลับไฟล์ .mid ได้ด้วยฟังก์ชันวินโดว์ API ที่ชื่อ mciSendString โดยในรูทีน PlayMIDI จะทำหน้าที่เล่นกลับไฟล์ .mid ที่กำหนดให้กับพารามิเตอร์ MIDIfilename$

```
Declare Function mciSendString Lib
  "mmsystem" (ByVal lpstrCommand$, ByVal
  lpstrReturnStr As Any, ByVal
  wReturnLen%, ByVal hCallBack%) As Long

Sub PlayMIDI (MIDIfilename$)
  Dim Dummy%, cmdstr$, devID$

  devID$ = "SEEDMIDI"
  cmdstr$ = "open " & MIDIfilename$ &
  type sequencer alias " & devID$
  Dummy% = mciSendString(cmdstr$, 0&, 0)
  cmdstr$ = "play " & devID$
  ' replace above line with underneath line
  ' for wait until play finish.
  ' cmdstr$ = "play " & devID$ & " wait"
  Dummy% = mciSendString(cmdstr$, 0&, 0)
  Dummy% = mciSendString("close
           Animation", 0&, 0, 0)
End Sub
```

การคัดลอกไอคอนลงในคลิปบอร์ด

ผู้อ่านไม่สามารถทำการคัดลอกไอคอนลงในคลิปบอร์ดของวินโดว์โดยตรง เนื่องจากคลิปบอร์ดไม่สนับสนุนกราฟิกในรูปแบบไอคอน ดังนั้นผู้อ่านต้องทำการแปลงรูปแบบของกราฟิกให้เป็นบิตแม็พเสียก่อน แล้วจึงคัดลอกกราฟิกในรูปแบบบิตแม็พลงในคลิปบอร์ด ดังนี้

โค้ดบรรทัดนี้จะเกิดข้อผิดพลาด "Invalid Clipboard fomat"

```
Clipboard.SetData LoadPicture
  ("c:\vb40\icons\comm\net01.ico")
```

โค้ดต่อไปนี้จะไม่ทำให้เกิดข้อผิดพลาด

```
Picture1.Picture = LoadPicture
  ("c:\vb40\icons\comm\net01.ico")
Picture1.Picture = Picture1.Image
Clipboard.SetData Picture1.Picture, 2
```

หมายเหตุ เราใช้ความสามารถของคอนโทรล picture box ในการแปลงรูปแบบกราฟิก

สี RGB เทียบเท่าฟังก์ชัน QBColor

ฟังก์ชัน QBColor สามารถแสดงผลได้ 16 สี (0 ถึง 15) ซึ่งเป็นสีพื้นฐานที่คอมแพตทิเบิลกับสีที่ใช้โดย QuickBASIC สำหรับรหัสสี R,G,B สำหรับฟังก์ชัน RGB ที่เทียบเท่ากับทั้ง 16 สี มีดังต่อไปนี้

QBColor	Red	Green	Blue
0-Black	&H00	&H00	&H00
1-Blue	&H00	&H00	&H80
2-Green	&H00	&H80	&H00
3-Cyan	&H00	&H80	&H80
4-Red	&H80	&H00	&H00
5-Magenta	&H80	&H00	&H80
6-Yellow	&H80	&H80	&H00
7-White	&HC0	&HC0	&HC0
8-Gray	&H80	&H80	&H80
9-Light Blue	&H00	&H00	&HFF
10-Light Green	&H00	&HFF	&H00
11-Light Cyan	&H00	&HFF	&HFF
12-Light Red	&HFF	&H00	&H00
13-Light Magenta	&HFF	&H00	&HFF
14-Light Yellow	&HFF	&HFF	&H00
15-Bright White	&HFF	&HFF	&HFF

ข้อควรจำเกี่ยวกับฟอร์ม MDI

ในการสร้างและใช้งานฟอร์ม MDI ด้วย VB/Win สิ่งที่ผู้อ่านควรจดจำและทำความเข้าใจเกี่ยวกับการควบคุมฟอร์ม MDI มีดังต่อไปนี้

- ฟอร์ม MDI ที่แสดงผลแบบ maximized ชื่อของฟอร์ม (caption) จะถูกนำมารวมกับชื่อของฟอร์มแม่และแสดงผลในแถบ title

- ฟอร์ม MDI ไม่สามารถเป็นฟอร์มชนิด modal

- ถ้าหากผู้อ่านไม่มีการกำหนดขนาดของฟอร์มในขณะที่โหลดด้วยวิธี เช่น Move เป็นต้น ขนาดของฟอร์มก็จะถูกควบคุมด้วยวินโดว์โดยตรง

- ถ้าหากมีการอ้างอิงถึงคอนโทรลหรือฟอร์มลูก MDI โดยตรง ฟอร์มแม่ MDI ก็จะถูกโหลดโดยอัตโนมัติและตามด้วยการโหลดฟอร์มลูก MDI

วิธีการค้นหาคอนโทรลในฟอร์ม MDI

สำหรับ VB/Win 4.0 คอนโทรลจะถูกจัดเก็บรวมเอาไว้ภายใน collection ที่ชื่อ Controls ดังผู้อ่านสามารถค้นหาคอนโทรลที่ต้องการภายในฟอร์ม MDI ได้อย่างรวดเร็วและถูกต้องที่สุด โดยอาศัยประโยคตัวอย่างต่อไปนี้

```
For Each Control In MDIForm1.Controls
  If Control.Name = "Supplier" Then
    ...
  End If
Next
```

สาเหตุการเกิด cascade สำหรับโพรซีเดอร์เหตุการณ์ Paint

โพรซีเดอร์เหตุการณ์ Paint เกิดขึ้นเมื่อพื้นที่บางส่วนหรือทั้งหมดของฟอร์มหรือคอนโทรล picture box ถูกทับหรือยื่นออกนอกขอบเขตแสดงผลของจอภาพ ดังนั้นภายในโพรซีเดอร์เหตุการณ์ Paint จึงไม่ควรมีโค้ดดังต่อไปนี้ เพื่อป้องกันการเกิดเหตุการณ์ cascade ซึ่งจะทำให้โปรแกรมหยุดการทำงานได้ทันที ดังนี้

- โค้ดสำหรับการย้ายหรือเปลี่ยนของฟอร์มหรือคอนโทรล เช่น Move หรือ Top เป็นต้น

- เปลี่ยนสีของฟอร์มหรือคอนโทรล เช่น การกำหนดค่าให้กับคุณสมบัติ ForeColor เป็นต้น

- ใช้วิธี Refresh

การกลับทิศทางภาพกราฟิก

VB/Win 4.0 สนับสนุนวิธี PaintPicture ซึ่งเป็นวิธีของ Form, Printer และ PictureBox สำหรับการวาดกราฟิก ซึ่งผู้อ่านสามารถใช้วิธีดังกล่าวในการกลับทิศทาง (flip) ภาพกราฟิกได้โดยใช้หลักการดังนี้

รูปแบบการใช้

```
object.PaintPicture picture, x1, y1,
       width1, height1, x2, y2, width2,
       height2, opcode
```

กลับภาพซ้าย-ขวา

กลับค่าของ width1 ให้มีค่าเป็นลบ

กลับภาพขึ้น-ลง

กลับค่าของ height1 ให้มีค่าเป็นลบ

ฟังก์ชันสำหรับด้านการเงิน

ตั้งแต่ VB/Win 3.0 เป็นต้นมา ไมโครซอฟท์ได้เตรียมกลุ่มของฟังก์ชันที่คำนวณด้านการเงินไว้หลายฟังก์ชัน เพื่อช่วยลดภาระการเขียนโปรแกรมสำหรับงานที่สำคัญ ๆ ด้านการเงิน ซึ่งสามารถแยกออกเป็นกลุ่ม ๆ ตามหน้าที่ได้ดังนี้

กลุ่ม	ฟังก์ชัน
Depreciation	DDB, SLN, SYN
Future value	FV
Payment	IPmt, Pmt, PPmt
Number of periods	NPer
Internal rate of return	IRR, MIRR
Present value	NPV, PV
Interest rate	Rate

การตั้งค่า TabIndex อย่างรวดเร็ว

ในการออกแบบฟอร์ม ในบางครั้งผู้อ่านอาจจะมีการแทรกคอนโทรลลงในฟอร์ม ซึ่งอาจทำให้ลำดับการรับโฟกัสของคอนโทรลเมื่อเลื่อนด้วยคีย์ TAB ไม่ต่อเนื่องตามลำดับตำแหน่งของคอนโทรล ซึ่งในกรณีนี้ผู้อ่านสามารถปรับลำดับการรับโฟกัสได้โดยการตั้งค่าของ TabIndex ใหม่ ซึ่งในกรณีนี้ผู้เขียนแนะนำให้ผู้อ่านใช้เทคนิคดังนี้ ให้เลือกทีละคอนโทรล โดยเริ่มต้นจากคอนโทรลสุดท้ายมายังคอนโทรลแรกสุด และในแต่ละครั้งที่เลือกคอนโทรล ผู้อ่านต้องกำหนดให้ค่าของ TabIndex เท่ากับ 0 ทุกครั้ง เมื่อสิ้นสุดการตั้งค่าดังกล่าว VB/Win ก็จะเรียงลำดับค่าของ TabIndex ให้ใหม่โดยอัตโนมัติ

การเปรียบเทียบสตริงที่เร็วที่สุด

สำหรับ VB/Win การเปรียบเทียบสตริงเป็นขบวนการที่ช้าที่สุด โดยเฉพาะเมื่อกระทำภายในวงลูป เพื่อเปรียบเทียบสตริงทีละค่า สำหรับตัวอย่างต่อไปนี้จะเป็นวิธีที่เร็วที่สุดในการเปรียบเทียบสตริงภายในวงลูป

```
Function SearchText(searchStr$) As Integer
 Dim i%

 For i% = 1 To UBound(strVar$)
     If Len(strVar$(i%)) =
                 Len(searchStr$) Then
        If strVar$(i%) = searchStr$
                        Then
            SearchText = i%
            Exit Function
        End If
     End If
 Next i%
End Function
```

ByVal กับพารามิเตอร์ชนิดสตริง

พารามิเตอร์ของฟังก์ชันวินโดว์ API ที่ต้องการข้อมูลชนิดสตริง โดยทั้งหมดต้องการสตริงในรูปแบบของภาษาซี หรือที่เรียกว่า Null-terminate string ดังนั้นในการกำหนดสตริงของ VB/Win ให้กับพารามิเตอร์เหล่านี้ผู้อ่านต้องทำการแปลงรูปแบบของข้อมูลสตริงเสียก่อนทุกครั้ง โดยการกำหนด ByVal ให้กับพารามิเตอร์ข้อมูลสตริงภานในประโยค Declare ทุกครั้ง เช่น

```
Declare Function FindAtom Lib "Kernel"
        (ByVal lpString As String) As Integer
```

ชนิดข้อมูลของภาษา C/C++

รายการต่อไปนี้เป็นชนิดข้อมูลที่สนับสนุนโดยภาษา C หรือ C++ และเป็นชนิดข้อมูลที่ถูกกำหนดให้กับพารามิเตอร์ต่าง ๆ ของฟังก์ชันวินโดว์ API ซึ่งผู้อ่านสามารถใช้ชนิดของข้อมูลที่สนับสนุนโดย VB/Win ที่มีขนาดจำนวนไบต์และความละเอียดที่ครอบคลุมค่าไว้ได้ มาใช้กำหนดให้กับพารามิเตอร์ต่าง ๆ ได้อย่างถูกต้อง

ชนิดข้อมูล	รายละเอียด
BOOL	บูลีนขนาด 16 บิต
BYTE	เลขจำนวนเต็ม unsigned integer ขนาด 8 บิต
char	เลขจำนวนเต็ม signed integer ขนาด 8 บิต
FARPROC	พอยน์เตอร์ชนิดไกลขนาด 32 บิต ชี้ไปยังรูทีน
HANDLE	ค่าหมายเลข handle ขนาด 16 บิต ของวินโดว์
int, short	เลขจำนวนเต็ม signed integer ขนาด 16 บิต
long	เลขจำนวนเต็ม signed integer ขนาด 32 บิต
LPINT	พอยน์เตอร์ชนิดไกลขนาด 32 บิต ชี้ไปยังข้อมูลชนิด signed integer
LPSTR	พอยน์เตอร์ชนิดไกลขนาด 32 บิต ชี้ไปยังสตริงรูปแบบภาษาซี
NPSTR	พอยน์เตอร์ชนิดใกล้ขนาด 16 บิต ชี้ไปยังสตริงรูปแบบภาษาซี
WORD	เลขจำนวนเต็ม unsigned integer ขนาด 16 บิต
DWORD	เลขจำนวนเต็ม unsigned integer ขนาด 16 บิต

ฟังก์ชันวินโดว์ API GetObject

ฟังก์ชันวินโดว์ API ที่ชื่อ GetObject เป็นฟังก์ชันที่ถูกใช้งานบ่อยครั้งที่สุด ในการเขียนโปรแกรมด้วยฟังก์ชันวินโดว์ API และเป็นฟังก์ชันที่มีชื่อเหมือนกับฟังก์ชันของ VB/Win 3.0 และ 4.0 ซึ่งเป็นฟังก์ชันสำหรับการอ้างถึงออบเจ็กต์ OLE Automation จากไฟล์ ดังนั้นเพื่อไม่ให้เกิดข้อผิดพลาด ผู้เขียนขอแนะนำให้ผู้อ่านประกาศฟังก์ชันวินโดว์ API ที่ชื่อ GetObject ด้วยชื่ออื่น ๆ ด้วยคำสงวน Alias ดังนี้

```
Declare Function GetObjectGDI Lib "GDI"
  Alias "GetObject" (ByVal hObject As
  Integer, ByVal nCount As Integer,
  ByVal lpObject As Long) As Integer
```

กลุ่มฟังก์ชันเปรียบเทียบตัวอักษร

วินโดว์มีกลุ่มของฟังก์ชันที่ช่วยในการเปรียบเทียบตัวอักษรว่าเป็นตัวเลขหรือตัวอักษร หรือเปรียบเทียบตัวอักษรพิมพ์เล็กหรือพิมพ์ใหญ่ โดยในรูทีน CheckCharIs จะรายงานข้อมูลสตริงบอกชนิดของตัวอักษร

```
Declare Function IsCharAlpha Lib "User"
      (ByVal cChar As Integer) As Integer
Declare Function IsCharAlphaNumeric Lib
      "User" (ByVal cChar As Integer) As
      Integer
Declare Function IsCharLower Lib "User"
      (ByVal cChar As Integer) As Integer
Declare Function IsCharUpper Lib "User"
      (ByVal cChar As Integer) As Integer

Sub CheckCharIs(char$)
      Dim ChrAsc%

      ChrAsc% = Asc(char$)
      If IsCharAlpha(ChrAsc%) Then
            Print char$; " is letter"
End If
If IsCharAlphaNumeric(ChrAsc%) Then
      Print char$; " is character or numeral" →
      End If
      If IsCharLower(ChrAsc%) Then
        Print char$; " is lower-case character"
      End If
      If IsCharUpper(ChrAsc%) Then
        Print char$; " is upper-case character"
      End If
End Sub ■
```

การใช้ประโยค Property Let

ประโยค Property Let ใช้ในการกำหนดค่าให้กับคุณสมบัติของคลาส ดังนั้นทุกครั้งที่มีการเรียกใช้ประโยค Property Let ผู้อ่านจำเป็นต้องกำหนดอาร์กิวเมนต์ให้กับประโยคดังกล่าวอย่างน้อย 1 ตัว เพื่อใช้เป็นค่าที่กำหนดให้กับคุณสมบัตินั้น ๆ เช่น ประโยคข้างกำหนดให้คุณสมบัติ RoomID ของคลาสให้มีค่าเท่ากับ F144 เป็นต้น

```
RoomID() = "F144"
```

รูปแบบการอ้างคุณสมบัติของ DAO

คุณสมบัติของ DAO นั้นมี 2 ชนิด คือ ชนิดที่มาพร้อมกับ DAO (built-in) และชนิดที่สร้างโดยผู้ใช้ (user-defined) ซึ่งรูปแบบในการอ้างอิงถึงคุณสมบัติทั้งสองชนิดนี้มีดังต่อไปนี้

ชนิดที่มาพร้อมกับ DAO

```
object.Properties("name")
object.Properties!name
object.name
object.Properties(number)
```

ชนิดที่สร้างโดยผู้ใช้

```
object.name
object.Properties(number)
```

ประโยค MsgBox กับโพรซีเดอร์เหตุการณ์ Scroll

โพรซีเดอร์เหตุการณ์ Scroll จะเกิดขึ้นเมื่อผู้ใช้มีการเลื่อนตำแหน่งปุ่มเลื่อนในแถบเลื่อน สำหรับโค้ดภายในโพรซีเดอร์เหตุการณ์นี้ ทางไมโครซอฟท์แนะนำว่าไม่ควรใช้ประโยค MsgBox ภายในโพรซีเดอร์เหตุการณ์นี้ เพราะสามารถทำให้เกิดข้อผิดพลาดได้

ฟังก์ชันทางคณิตศาสตร์ตรีโกณ

ฟังก์ชันต่อไปนี้เป็นฟังก์ชันช่วยการคำนวณด้านคณิตศาสตร์ตรีโกณมิติ

ชื่อฟังก์ชัน	รายละเอียด
ArcSin	Inverse Sine
ArcCos	Inverse Cosine
ArcTan	Inverse Tangent
HypSin	Hyperbolic Sine
HypCos	Hyperbolic Cosine
HypTan	Hyperbolic Tangent
HypArcSin	Inverse Hyperbolic Sine
HypArcCos	Inverse Hyperbolic Cosine
HypArcTan	Inverse Hyperbolic Tangent

```
Function ArcSin(X As Single) As Single
      ArcSin = Atn(X / Sqr(-X * X + 1))
End Function
Function ArcCos(X As Single) As Single
      ArcCos = Atn(-X / Sqr(-X * X + 1))
                  + 2 * Atn(1)
End Function
Function ArcTan(X As Single) As Single
      ArcTan = ArcSin(X) / ArcCos(X)
End Function
Function HypSin(X As Single) As Single
      HypSin = (Exp(X) - Exp(-X)) / 2
End Function
Function HypCos(X As Single) As Single
      HypCos = (Exp(X) + Exp(-X)) / 2
End Function
Function HypTan(X As Single) As Single
     ' HypTan = HypSin(X) / HypCos(X)
End Function
Function HypArcSin(X As Single) As Single
      HypArcSin = Log(X + Sqr(X * X + 1))
End Function
Function HypArcCos(X As Single) As Single
      HypArcCos = Log(X + Sqr(X * X - 1))
End Function
Function HypArcTan(X As Single) As Single
      HypArcTan = HypArcSin(X) / HypArcCos(X)
End Function
```

การเลื่อนโฟกัสด้วยโค้ด

โดยปกติในขณะรันแอพพลิเคชัน การเปลี่ยนโฟกัสจากคอนโทรลหนึ่งไปยังอีกคอนโทรลหนึ่งจะเกิดขึ้นเมื่อผู้ใช้กดคีย์ TAB หรือใช้เม้าส์คลิ๊กเลือกคอนโทรลอื่น ๆ ที่ต้องการ แต่ผู้อ่านสามารถเปลี่ยนโฟกัสจากคอนโทรลหนึ่งไปยังคอนโทรลถัดไปได้โดยใช้ฟังก์ชัน SendKeys ดังในตัวอย่างต่อไปนี้จะเลื่อนโฟกัสไปยังคอนโทรลถัดไปเมื่อผู้ใช้กรอกตัวอักษรให้กับคอนโทรล text box เท่ากับจำนวนที่กำหนดไว้โดยคุณสมบัติ MaxLength ดังนี้

```
Sub Text1_KeyPress(KeyAscii As Integer)
      If Len(Text1) = Text1.MaxLength Then
            SendKeys "{TAB}"
      End If
End Sub
```

สร้างคุณสมบัติ Undo ให้กับคอนโทรล Text Box

ผู้อ่านสามารถเรียกกลับข้อความทั้งหมดที่ถูกแก้ไขภายในคอนโทรล text box ได้โดยอาศัยฟังก์ชันวินโดว์ API ที่ชื่อ SendMessage โดยในรูทีน UndoTextBox จะทำหน้าที่ดังกล่าว

```
Declare Function SendMessage Lib "User"
      (ByVal hWnd As Integer, ByVal wMsg As
      Integer, ByVal wParam As Integer,
      lParam As Any) As Long
Const WM_USER = &H400
Const EM_UNDO = WM_USER + 23

Sub UndoTextBox(TB As TextBox)
      Dim Dummy&
      Dummy& = SendMessage(TB.hWnd,
                        EM_UNDO, 0, 0&)
End Sub
```

หมายเหตุ ถ้าหากคอนโทรลที่ส่งให้กับรูทีน UndoTextBox ไม่ใช่ text box ก็จะเกิดข้อผิดพลาดทันที

การเรียกโพรซีเดอร์เหตุการณ์ Click

คอนโทรลส่วนใหญ่ของ VB/Win สามารถเกิดเหตุการณ์ Click ได้ ซึ่งเหตุการณ์นี้สามารถเกิดขึ้นได้โดยอาศัยโค้ดนั้นมีอยู่ด้วยกัน 2 วิธี ดังนี้ โดยใช้คุณสมบัติ Value เช่น

```
cmdOK.Value = True
```

หรือโดยการเรียกชื่อโพรซีเดอร์โดยตรง เช่น

```
cmdOK_Click
```

แต่อย่างไรก็ตาม ผู้เขียนขอแนะนำให้ผู้อ่านใช้วิธีการเรียกชื่อโพรซีเดอร์โดยตรง ถ้าหากโค้ดอยู่ภายในฟอร์มที่เป็นเจ้าของคอนโทรลนั้น ๆ เพราะการเรียกชื่อโพรซีเดอร์โดยตรง จะเท่ากับการเรียกใช้รูทีนใด ๆ ของ VB/Win ซึ่งจะส่งผลให้โปรแกรมสามารถปฏิบัติโค้ดได้เร็วกว่าวิธีแรก แต่วิธีนี้ไม่สามารถนำไปใช้ได้กับโค้ดภายในโมดูลหรือฟอร์มอื่น ๆ ที่ไม่ได้เป็นเจ้าของคอนโทรล